

객체지향 방법론 중 Jaconson 방법

Jacobson 방법론은 Ivar Jacobson이 개발한 객체지향 분석 및 설계(Object-Oriented Analysis and Design, OOAD) 방법론으로, 특히 ****유스케이스(Use Case)****를 중심으로 시스템을 모델링하는 데 중점을 둡니다. 이 방법론은 소프트웨어 개발의 전체 라이프사이클을 다루며, 추적 가능성과 재사용성을 강조합니다. Jacobson의 방법론은 크게 ****OOSE(Object-Oriented Software Engineering)****와 **Objectory**라는 두 가지로 알려져 있습니다.

Jacobson 방법론의 주요 특징

1. 유스케이스 중심 접근법:

- Jacobson 방법론의 핵심은 ****유스케이스(Use Case)****입니다.
- 유스케이스는 사용자와 시스템 간의 상호작용을 설명하며, 시스템 요구사항을 명확히 정의하는 데 사용됩니다.
- 유스케이스는 분석, 설계, 검증, 테스트 등 소프트웨어 개발 프로세스 전반에 걸쳐 활용됩니다 ^{[1] [2] [3]}.

2. 전체 라이프사이클 지원:

- 요구사항 수집부터 설계, 구현, 테스트까지 소프트웨어 개발의 모든 단계를 포괄합니다.
- 각 단계 간의 추적 가능성을 보장하여 분석 및 설계 작업의 재사용성을 높입니다 ^{[1] [4]}.

3. 모델 기반 접근:

- Objectory(객체 공장)라는 이름으로 알려진 Jacobson의 방법론은 여러 모델로 구성됩니다:
 - **유스케이스 모델:** 시스템 내부와 외부 간의 상호작용 정의.
 - **도메인 객체 모델:** 실제 세계 객체를 시스템 내 객체로 매핑.
 - **분석 객체 모델:** 소스 코드로 구현될 객체 구조 정의.
 - **구현 모델:** 시스템 구현을 위한 세부사항 포함.
 - **테스트 모델:** 테스트 계획, 사양, 보고서를 포함 ^{[1] [2] [5]}.

4. 산업화된 소프트웨어 개발:

- Objectory는 대규모 실시간 시스템 개발을 목표로 설계되었으며, 유지보수성과 확장성을 강조합니다 ^[5].

Jacobson 방법론의 주요 구성 요소

1. 유스케이스 모델 (Use Case Model):

- 유스케이스는 시스템과 외부 사용자(액터) 간의 상호작용을 설명합니다.
- 유스케이스는 다음과 같은 정보를 포함해야 합니다:
 - 유스케이스가 시작하고 끝나는 시점.
 - 액터와 시스템 간의 상호작용 내용.
 - 데이터 저장 및 조회 방식.
 - 예외 상황 처리 흐름^{[1] [2] [4]}.

2. 도메인 객체 모델 (Domain Object Model):

- 실제 세계의 객체를 시스템 내에서 사용할 수 있도록 매핑합니다.
- 객체 간 관계(예: 상속, 연관)를 정의합니다.

3. 분석 객체 모델 (Analysis Object Model):

- 소프트웨어 설계를 위한 구조적 분석을 수행합니다.
- 클래스 다이어그램 및 상태 전이 다이어그램 등을 포함합니다.

4. 구현 모델 (Implementation Model):

- 분석 결과를 바탕으로 실제 코드로 구현할 수 있는 세부 사항을 정의합니다.

5. 테스트 모델 (Test Model):

- 테스트 계획과 사양을 작성하고 결과를 문서화합니다.

Jacobson 방법론의 장점

1. 유스케이스 기반 설계로 사용자 요구사항 반영:

- 유스케이스는 사용자 관점에서 시스템 동작을 설명하므로 요구사항 수집과 검증에 효과적입니다.^{[1] [3]}.

2. 추적 가능성과 재사용성:

- 각 단계 간의 추적 가능성을 보장하여 분석 및 설계 작업이 재사용될 수 있습니다.^[4].

3. 대규모 시스템에 적합:

- Objectory는 대규모 실시간 시스템 개발에 적합하도록 설계되었습니다.^[5].

Jacobson 방법론과 UML

- Jacobson 방법론은 UML(Unified Modeling Language)의 발전에 큰 영향을 미쳤습니다.
- 특히 유스케이스 다이어그램은 UML에서 표준화된 다이어그램 유형으로 채택되었습니다.

요약

Jacobson 방법론은 유스케이스를 중심으로 한 객체지향 분석 및 설계 방법론으로, 소프트웨어 개발 과정에서 요구사항 추적과 재사용성을 강조합니다. 이 접근법은 특히 대규모 산업용 소프트웨어 개발에 적합하며, 현대적인 OOAD 기법과 UML 발전에 중요한 기여를 했습니다.

✻

1. <https://bcanotes4u171863936.wordpress.com/jacobson-methodology/>
2. <https://www.studocu.com/in/document/jawaharlal-nehru-technological-university-kakinada/object-oriented-analysis-and-design/ooad6/41890479>
3. <https://iq.opengenus.org/rumbaugh-booch-and-jacobson-methodologies/>
4. <https://www.slideserve.com/aromer/the-jacobson-et-al-methodology-powerpoint-ppt-presentation>
5. <https://www.ivarjacobson.com/publications/books/object-oriented-software-engineering-book>